Delay Optimization in a Simple Offloading System

Darin Jeff and Eytan Modiano
LIDS, Massachusetts Institute of Technology, Cambridge, MA, USA
Email: {djeff, modiano}@mit.edu

***Abstract*—We consider a computation offloading system where jobs are processed sequentially at a local server followed by a higher-capacity cloud server. The system offers two service modes, differing in how the processing is split between the servers. Our goal is to design an optimal policy for assigning jobs to service modes and partitioning server resources in order to minimize delay. We begin by characterizing the system's stability region and establishing design principles for service modes that maximize throughput. For any given job assignment strategy, we derive the optimal resource partitioning and present a closed-form expression for the resulting delay. Moreover, we establish that the delay-optimal assignment policy exhibits a distinct breakaway structure: at low system loads, it is optimal to route all jobs through a single service mode, whereas beyond a critical load threshold, jobs must be assigned across both modes. We conclude by validating these theoretical insights through numerical evaluation.**

***Index Terms*—computation offloading, delay-optimal control, delay-optimal system design, cloud computing.**

I. Introduction

The rapid adoption of Large Language Models (LLMs) such as GPT-4, Claude, and Mistral has significantly increased demand for cloud computing resources. These models require memory-intensive architectures that generally exceed the capabilities of consumer-grade hardware, leading most LLM-based applications to rely heavily on a centralized cloud infrastructure for inference and fine-tuning.

At the same time, end-user devices are becoming increasingly capable. Hardware manufacturers such as Apple, AMD, and Qualcomm have recently announced plans for consumer-grade devices that can run models with a few billion parameters locally. In parallel, a growing body of work [1]–[3] has explored strategies for partitioning LLM workloads across user and cloud devices. Together, these trends motivate a re-examination of conventional cloud-centric processing pipelines. Instead of the conventional strategy of offloading most of the computation to the cloud, systems can adopt alternative strategies that place a greater portion of the workload on user devices when appropriate, thereby improving overall system capacity and delay performance.

Recently, there has been a tremendous amount of interest in the topic of computational offloading. Qin et al. [4], [5] proposed distributed threshold-based offloading policies for mobile cloud computing systems, formulating queue-aware decision-making as a game-theoretic problem and establishing equilibrium properties. Similarly, Zhou et al. [6] utilized queue-length-based thresholds in multi-agent Mobile Edge Computing systems and demonstrated that their distributed best-response algorithm achieves near-optimal offloading utility compared to centralized benchmarks. Lyapunov-based control techniques have also been explored in distributed computing networks [7], [8], where drift-plus-penalty methods are used to design algorithms that are throughput-optimal while enabling tunable trade-offs between delay and cost in offloading settings. Finally, in closely related settings, the authors in [9] showed that delay-optimal queue-length-based policies exhibit a switch-type structure.

While prior work has focused on optimizing computation offloading in specific system models, our goal is to develop a more fundamental understanding of such systems. To that end, we study a two-stage offloading system in which each job is processed sequentially by a local server with limited capacity, followed by a more powerful cloud server. Jobs are served using one of two service modes, each specifying how the jobs workload is partitioned across the two servers. One mode is "cloud-heavy", offloading a larger fraction of computation, while the other is "local-heavy" and relies more heavily on local processing. This dual-mode structure captures systems that have traditionally favored cloud-centric execution but are increasingly capable of greater local processing due to advances in end-user hardware.

Remark (Model generality). Although our model is motivated by computational offloading, it is not inherently tied to computation. The two-stage structure also captures settings in which limited communication capacity, rather than processing power, is the primary system bottleneck. In such cases, the first stage can be interpreted as local preprocessing (e.g., compression or feature extraction), while the second stage represents transmission over a constrained communication link. The analysis and structural insights developed in this paper apply equally to these settings.

The primary goal of this paper is to characterize the optimal mode-assignment and server resource allocation strategy that stabilizes the system for all feasible arrival rates while minimizing average job delay.

We first characterize the stability region of the proposed dual-mode system and highlight key system design considerations, showing how poorly chosen service modes can introduce throughput bottlenecks. We then identify fundamental trade-offs between delay performance and achievable throughput. Finally, in the general case, we establish that the delay-optimal operating policy exhibits a simple yet intuitive breakaway structure: at low arrival rates, it is optimal to exclusively use the "cloud-heavy" mode, whereas beyond a critical threshold

arrival rate, assigning some jobs to "local-heavy" processing becomes necessary.

The remainder of the paper is organized as follows. Section II introduces the dual-mode system model. Section III develops preliminary analytical tools and results that underpin the subsequent analysis. In Section IV, we characterize the system's stability region and discuss key design implications. Section V derives the optimal allocation of server resources for a given mode assignment strategy. Section VI establishes structural properties of the optimal assignment policy. Section VII presents simulation results that validate the analytical findings. Finally, Section VIII concludes the paper.

II. System Model

In this section, we present the system model, which we call the *dual-mode* system. The system consists of two sequential servers - a local server followed by a cloud server. Incoming jobs are processed using one of two available service modes (SM). Each server maintains separate queues and partitions its computing resources to serve these modes independently. Jobs arrive following a Poisson process with rate λ. Upon arrival, each job is assigned to one of the two service modes: SM1 with probability p, or SM2 with probability $\bar{p} = 1 - p$. Each server reserves a fixed portion of its resources exclusively for each mode. Specifically, the fractions α and β represent the local and cloud resources allocated to SM1, with the complementary fractions $\bar{\alpha}$ and $\bar{\beta}$ allocated to SM2. Thus, the system's *operating point* is defined by the *assignment parameter* p and the *partition parameters* α and β.

We denote by μ_{l1} and μ_{c1} the service rates at the local and cloud servers, respectively, when their entire capacities are dedicated to SM1 jobs. Similarly, μ_{l2} and μ_{c2} represent full-capacity service rates for SM2. Without loss of generality, we designate SM1 as the "cloud-heavy" mode, relying more heavily on cloud processing. Correspondingly, SM2 is the "local-heavy" mode, with a greater share of computation performed at the local server. Formally, this is represented by:

$$\mu_{c1} < \mu_{c2} \quad \text{and} \quad \mu_{l2} < \mu_{l1}$$

At a given operating point (p, α, β), SM1 jobs experience independent, exponentially distributed service times with rates $\alpha\mu_{l1}$ and $\beta\mu_{c1}$ at the local and cloud servers, respectively. Similarly, SM2 jobs are served at rates $\bar{\alpha}\mu_{l2}$ and $\bar{\beta}\mu_{c2}$. This is illustrated in Figure 1.

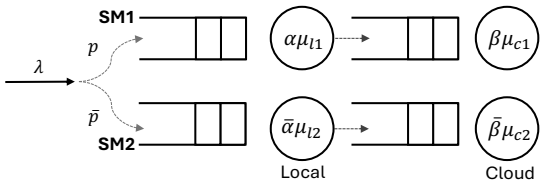


Fig. 1. Dual-Mode System depicting job arrivals, probabilistic service mode assignment, and dedicated resource allocation at the servers.

The objective is to determine the optimal operating point that minimizes the average delay experienced by jobs.

A. Canonical Transformation

Up to this point, we have described the system using explicit service rates associated with each mode. While directly observable, this parameterization can obscure underlying structural insights. To address this, we introduce a canonical transformation that maps the rate parameters into parameters that capture (i) the effective capacity of the local server, (ii) the relative capacity of the cloud server compared to the local server, and (iii) the division of workload across service modes.

Define

$$\mu_0 \coloneqq \frac{\mu_{l1}\mu_{l2}(\mu_{c2} - \mu_{c1})}{\mu_{l1}\mu_{c2} - \mu_{l2}\mu_{c1}} \qquad K \coloneqq \frac{\mu_{c1}\mu_{c2}(\mu_{l1} - \mu_{l2})}{\mu_{l1}\mu_{l2}(\mu_{c2} - \mu_{c1})}$$
$$f_1 \coloneqq \frac{\mu_{l2}(\mu_{c2} - \mu_{c1})}{\mu_{l1}\mu_{c2} - \mu_{l2}\mu_{c1}} \qquad f_2 \coloneqq \frac{\mu_{l1}(\mu_{c2} - \mu_{c1})}{\mu_{l1}\mu_{c2} - \mu_{l2}\mu_{c1}}$$

Here, μ_0 and $K\mu_0$ capture the effective processing capacities of the local and cloud servers, respectively. The parameters f_1 and f_2 represent the fraction of total job workload executed locally in service modes SM1 and SM2. Under this interpretation, a job's workload is divided between local and cloud processing according to the ratio $f_i : (1 - f_i)$ for mode i. The original service rates can be uniquely recovered from this representation using the following mapping:

$$\mu_{l1} = \frac{\mu_0}{f_1}, \quad \mu_{l2} = \frac{\mu_0}{f_2}, \quad \mu_{c1} = \frac{K\mu_0}{1 - f_1}, \quad \mu_{c2} = \frac{K\mu_0}{1 - f_2}$$

Our analysis later shows that the order of servers does not affect the system's stability region or delay performance. Thus, without loss of generality, we take $K > 1$, indicating that the cloud server has greater processing capacity compared to the local server. In the following section, we conduct preliminary analysis and develop essential analytical tools to study this system.

III. Preliminary Analysis

In the previous section, we introduced a canonical representation for the dual-mode system that explicitly captures server capacities and workload distribution. Extending this perspective - where a service mode defines how job load is distributed across servers - we now consider an offloading system with a single, adjustable service mode, which we call the *tunable-mode system*. We later demonstrate that this simpler system provides a useful benchmark for evaluating the stability and delay performance of the dual-mode system with comparable server capacities.

A. Tunable-Mode System

The tunable-mode system comprises two sequential servers: a local server with capacity μ_0 and a cloud server with higher capacity $K\mu_0$, where $K > 1$. Jobs arrive according to a Poisson process with rate λ.

Each incoming job is processed under the same service strategy, characterized by a tunable parameter $f \in [0, 1]$, which

determines the fraction of processing allocated to the local server. This parameter is fixed during system operation, but can be optimized offline based on system characteristics. Each server maintains a queue for incoming tasks, as depicted in Figure 2.

Under *service fraction parameter* f, the processing times at each server follow independent exponential distributions with service rates μ_0/f at the local server and $K\mu_0/(1-f)$ at the cloud server. This ensures a constant expected total processing requirement per job, independent of f.

The system objective is to select the optimal service fraction parameter, $f \in [0,1]$ under which the average delay in the system is minimized.

B. Delay Performance

The single-mode configuration evolves as a standard two-node Jackson tandem network, which is well-established in classical queueing theory (see, e.g., [10]). Classical results establish that stability is impossible if the arrival rate λ equals or exceeds the combined system capacity μ^*, defined as:

$$\mu^* = (K+1)\mu_0.$$

We define the stability region Λ_{TM} (mnemonically tunable-mode) as:

$$\Lambda_{\text{TM}} = \{\lambda > 0 \,|\, \lambda < \mu^*\}$$

For arrival rates $\lambda \in \Lambda_{\text{TM}}$, the system remains stable if the *service-fraction* parameter f lies within the set $\mathcal{F}_\lambda$, defined as:

$$\mathcal{F}_\lambda \coloneqq [0,1] \cap \left(1 - \frac{K\mu_0}{\lambda}, \frac{\mu_0}{\lambda}\right).$$

The expected delay in this system is given by:

$$T_{\text{TM}}(f;\lambda) \coloneqq \frac{f}{\mu_0 - f\lambda} + \frac{\bar{f}}{K\mu_0 - \bar{f}\lambda}$$

which captures both the wait and processing times.

We characterize the optimal service fraction parameter in the following theorem.

Theorem 1. For arrival rates $\lambda \in \Lambda_{\text{TM}}$, the delay-optimal service fraction parameter is:

$$f^*(\lambda) = \max\{f_{\min}(\lambda), 0\}$$

where

$$f_{\min}(\lambda) \coloneqq \frac{\lambda - \mu_0(K - \sqrt{K})}{\lambda(1+\sqrt{K})}$$

The corresponding average delay under $f^*(\lambda)$ is:

$$T^*_{\text{TM}}(\lambda) \coloneqq T_{\text{TM}}(f^*(\lambda);\lambda) = \begin{cases} \frac{2\lambda - (\sqrt{K}-1)^2\mu_0}{\lambda((K+1)\mu_0 - \lambda)}, & \lambda > (K-\sqrt{K})\mu_0 \\ \frac{1}{K\mu_0 - \lambda}, & \lambda \le (K-\sqrt{K})\mu_0 \end{cases}$$

Proof: Proof deferred to the technical report [**?**]. ∎

Theorem 1 shows that at lower arrival rates $\left(\lambda \le (K-\sqrt{K})\mu_0\right)$, the optimal service strategy assigns all processing exclusively to the cloud server (setting $f = 0$). As the arrival rate increases beyond this threshold, it becomes beneficial to distribute the workload between local and cloud servers. Moreover, the optimal fraction allocated locally, $f^*(\lambda)$ always remains below $1/(K+1)$ – which is the relative capacity of the local server to the total system capacity.

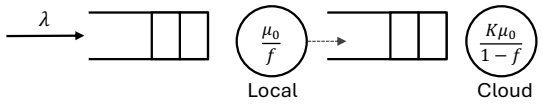


Fig. 2. Tunable-Mode System depicting job arrivals, and service times at each server.

IV. Stability Analysis

In this section, we return to the original system model, leveraging results from our previous analysis. We analyze the stability region in terms of the original rate parameters and glean interpretive insights later using their canonical forms.

As described in Section II, we represent service rates under the two service modes by $\mu_{l1}, \mu_{l2}, \mu_{c1}, \mu_{c2}$, and their corresponding canonical representation by μ_0, K, f_1, f_2.

Although our analysis focuses on stationary randomized policies, classical network optimization literature [11] establishes that these policies fully characterize the stability region, even when dynamic, state-dependent policies are considered.

The characterization of the stability region naturally consists of two parts: an achievability argument, where we construct a stabilizing policy that supports all arrival rates within the region, and a converse, which shows that no policy can stabilize the system for rates outside this region. The stability region for the dual-mode system is formally stated below.

Theorem 2 (Stability Region). The stability region Λ_{DM} for the dual-mode system is:

$$\Lambda_{\text{DM}} \coloneqq \{\lambda \ge 0 : \lambda < \lambda_{\max}\}$$

where

$$\lambda_{\max} \coloneqq \min\{\mu_{l1}, \mu_{c2}, \mu^*\}$$

and

$$\mu^* \coloneqq \frac{\mu_{c1}\mu_{c2}(\mu_{l1} - \mu_{l2}) + \mu_{l1}\mu_{l2}(\mu_{c2} - \mu_{c1})}{\mu_{l1}\mu_{c2} - \mu_{l2}\mu_{c1}}$$

Proof: The result follows directly from Lemma 1 (achievability) and Lemma 2 (converse), which we state and prove next. ∎

Recall that in this system, incoming jobs are randomly divided into two subsystems associated with each service mode. Due to the Poisson splitting property, each subsystem behaves as an independent two-node Jackson tandem network with server capacities determined by parameters α and β, and arrival rate governed by parameter p, as illustrated in Figure 3.

From standard stability criteria in queueing theory, the dual-mode system is stable if and only if the arrival rates into each server are strictly less than their corresponding service rates. Formally, the system is stable if and only if there exist parameters $p, \alpha, \beta \in [0,1]$ satisfying:

$$\lambda p < \alpha \mu_{l1}, \quad \lambda \bar{p} < \bar{\alpha} \mu_{l2}, \quad \lambda p < \beta \mu_{c1}, \quad \lambda \bar{p} < \bar{\beta} \mu_{c2} \quad (1)$$

Lemma 1 (Achievability). For the dual-mode system, if $\lambda \in \Lambda_{\text{DM}}$, then there exist $p, \alpha, \beta \in [0,1]$ such that the inequalities in (1) are satisfied.

Proof: The result follows by explicitly constructing values of p, α, and β that satisfy the inequalities in (1). Details are provided in the technical report [**?**]. ■

Lemma 2 (Converse). For the dual-mode system, if $\lambda \notin \Lambda_{\text{DM}}$, then, the inequalities in (1) are not satisfied for any $p, \alpha, \beta \in [0,1]$.

Proof: The result follows via a contradiction argument. Details are provided in the technical report [**?**]. ■

A. Interpretation

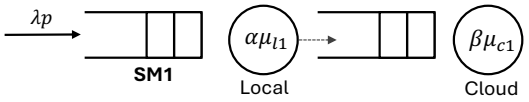
We offer some intuitive reasoning to clarify the structure of this stability region characterized by:

$$\lambda_{\max} = \min\{\mu_{l1}, \mu_{c2}, \mu^*\}$$

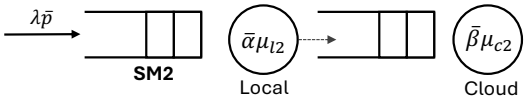
By convention, we have $\mu_{l1} > \mu_{l2}$ and $\mu_{c2} > \mu_{c1}$. Thus, μ_{l1} represents the maximum achievable processing rate at the local server, and μ_{c2} represents the maximum achievable processing rate at the cloud server. To interpret μ^*, we express it in terms of the canonical parameters:

$$\mu^* = (K+1)\mu_0$$

Recall, this was the same definition for μ^* in the tunable-mode system and represents the combined capacity of the local and cloud server.



(a) Service Mode 1 subsystem



(b) Service Mode 2 subsystem

Fig. 3. Independent subsystems resulting from Poisson splitting in the dual-mode configuration.

B. System Design Considerations

In the single-mode configuration, we saw that a well-designed service mode can stabilize arrival rates up to $\lambda_{\max} = \mu^*$. In contrast, if $\lambda_{\max} < \mu^*$ in the dual-mode system, it indicates poor service mode design, with both modes heavily relying on either the local or cloud server, creating a bottleneck. Conversely, $\lambda_{\max} = \mu^*$ indicates a balanced design capable of fully utilizing both server capacities.

Formally, if we consider a system with $\mu_{l1} < \mu^*$, rewriting this condition in terms of the canonical parameters yields

$$\mu_{l1} < \mu^* \Leftrightarrow \frac{1}{K+1} < f_1 < f_2$$

Having both $f_1, f_2 > \frac{1}{K+1}$ means each service mode assigns workloads to the local server that exceed its proportional capacity μ_0 relative to the overall system capacity $(K+1)\mu_0$ causing a throughput bottleneck. Similarly, the case $\mu_{c2} < \mu^*$ gives:

$$\mu_{c2} < \mu^* \Leftrightarrow f_1 < f_2 < \frac{1}{K+1}$$

which corresponds to systems inadequately utilizing the cloud's capacity due to excessive local processing.

Since maximizing the stability region is an important consideration in designing service modes for offloading systems, we define systems achieving this as *throughput-efficient*:

Definition 1

A dual-mode system is *throughput efficient* if its canonical service mode parameters satisfy:

$$f_1 < \frac{1}{K+1} < f_2.$$

V. Optimal Resource Allocation

In the previous section, we established the existence of stabilizing operating points for arrival rates within the stability region. Here, we characterize the delay-optimal partition parameters – α^*, β^* for a given value of the assignment parameter – p. For this, we need to ensure that stabilizing partition parameters exist for this p. Following the stability analysis in Section IV, we see that stabilizing choices of partition parameters exist for arrival rate $\lambda \in \Lambda_{\text{DM}}$, when the assignment parameter satisfies:

$$p \in \mathcal{P}_\lambda := [0,1] \cap (p_{\min}, p_{\max})$$

where: $p_{\min} = 1 - \frac{1/\lambda - 1/\mu_{l1}}{1/\mu_{l2} - 1/\mu_{l1}}$ and $p_{\max} = \frac{1/\lambda - 1/\mu_{c2}}{1/\mu_{c1} - 1/\mu_{c2}}$.

From standard queueing theory, the average delay in a dual-mode system operating stably at rate $\lambda \in \Lambda_{\text{DM}}$ is given by:

$$T_{\text{DM}}(p, \alpha, \beta; \lambda) = \frac{p}{\alpha\mu_{l1} - \lambda p} + \frac{\bar{p}}{\bar{\alpha}\mu_{l2} - \lambda \bar{p}} + \frac{p}{\beta\mu_{c1} - \lambda p} + \frac{\bar{p}}{\bar{\beta}\mu_{c2} - \lambda \bar{p}}$$

We optimize this expression to obtain the optimal partition parameters.

Theorem 3 (Optimal Resource Allocation). For a dual-mode system operating at $\lambda \in \Lambda_{\text{DM}}$ with fixed assignment parameter $p \in \mathcal{P}_\lambda$, the delay-optimal server partition parameters are:

$$\alpha^*(p;\lambda) = \frac{\frac{\lambda p}{\mu_{l1}}\sqrt{\frac{\bar{p}}{\mu_{l2}}} + \left(1 - \frac{\lambda \bar{p}}{\mu_{l2}}\right)\sqrt{\frac{p}{\mu_{l1}}}}{\sqrt{\frac{p}{\mu_{l1}}} + \sqrt{\frac{\bar{p}}{\mu_{l2}}}}$$

and

$$\beta^*(p;\lambda) = \frac{\frac{\lambda p}{\mu_{c1}}\sqrt{\frac{\bar{p}}{\mu_{c2}}} + \left(1 - \frac{\lambda \bar{p}}{\mu_{c2}}\right)\sqrt{\frac{p}{\mu_{c1}}}}{\sqrt{\frac{p}{\mu_{c1}}} + \sqrt{\frac{\bar{p}}{\mu_{c2}}}}.$$

The corresponding delay under the optimal partitioning is:

$$T^*_{\text{DM}}(p;\lambda) = \frac{\left(\sqrt{\frac{p}{\mu_{l1}}} + \sqrt{\frac{\bar{p}}{\mu_{l2}}}\right)^2}{1 - \frac{\lambda p}{\mu_{l1}} - \frac{\lambda \bar{p}}{\mu_{l2}}} + \frac{\left(\sqrt{\frac{p}{\mu_{c1}}} + \sqrt{\frac{\bar{p}}{\mu_{c2}}}\right)^2}{1 - \frac{\lambda p}{\mu_{c1}} - \frac{\lambda \bar{p}}{\mu_{c2}}}$$

Proof: The result follows from elementary convex optimization. Details are provided in the technical report [**?**]. ■

A. Interpretation

We now provide intuition to better interpret the expression for the delay under the optimal partitioning. First, we re-write the expression using the canonical parameters:

$$T^*_{\text{DM}}(p;\lambda) = \frac{\left(\sqrt{pf_1} + \sqrt{\bar{p}f_2}\right)^2}{\mu_0 - \lambda\left(pf_1 + \bar{p}f_2\right)} + \frac{\left(\sqrt{p\bar{f}_1} + \sqrt{\bar{p}\bar{f}_2}\right)^2}{K\mu_0 - \lambda\left(p\bar{f}_1 + \bar{p}\bar{f}_2\right)}$$

Now, define $f(p) \coloneqq pf_1 + (1-p)f_2$, which represents the effective fraction of workload processed locally for a system operating under assignment parameter p.
Now, we can succinctly rewrite the optimal dual-mode delay as:

$$T^*_{\text{DM}}(p;\lambda) = T_{\text{TM}}(f(p);\lambda) + T_{\text{OH}}(p;\lambda)$$

where $T_{\text{TM}}(\cdot;\lambda)$ is the delay function from the tunable-mode system, given by:

$$T_{\text{TM}}(f;\lambda) \coloneqq \frac{f}{\mu_0 - f\lambda} + \frac{\bar{f}}{K\mu_0 - \bar{f}\lambda}$$

and $T_{\text{OH}}(\cdot;\lambda)$ denotes a non-negative overhead term:

$$T_{\text{OH}}(p;\lambda) \coloneqq \; 2\sqrt{p\bar{p}}\left\{\frac{\sqrt{f_1 f_2}}{\mu_0 - \lambda(pf_1 + \bar{p}f_2)} + \frac{\sqrt{\bar{f}_1\bar{f}_2}}{K\mu_0 - \lambda\left(p\bar{f}_1 + \bar{p}\bar{f}_2\right)}\right\}$$

which captures the additional delay incurred from allocating server resources across different modes to achieve a desired local-to-cloud workload split – $f(p)$.

This decomposition shows that the delay of a dual-mode system operating under assignment parameter p can be expressed as the delay of an equivalent tunable-mode system with service fraction $f(p)$, plus an additional overhead cost.

B. System Design Considerations

The delay decomposition in (V-A) provides a natural lower bound on the dual-mode system's delay:

Proposition 1. For any dual-mode system with fixed server capacities μ_0 and $K\mu_0$ (in canonical form), and operating at arrival rate $\lambda \in \Lambda_{\text{DM}}$, the delay under any feasible assignment parameter $p \in \mathcal{P}_\lambda$ is lower bounded by the optimal delay achievable in the corresponding tunable-mode system. Formally,

$$T^*_{\text{DM}}(p;\lambda) \geq T^*_{\text{TM}}(\lambda), \quad \forall\, p \in \mathcal{P}_\lambda,\ \lambda \in \Lambda_{\text{DM}}.$$

Proof: The result is immediate from (V-A) and the non-negativity of the overhead term. ■

Proposition 1 indicates that the optimal tunable-mode delay serves as a fundamental performance benchmark, independent of specific mode parameters (f_1 and f_2) and determined solely by server capacities and arrival rate.

In the next section, we build on the intuitive delay representation in (V-A) and our tunable-mode analysis to identify a class of systems in which one mode becomes redundant under optimal operation. We also show that in the general case, the optimal strategy exhibits a breakaway structure, which we characterize next.

VI. Optimal Assignment Strategy

Having identified optimal server allocations for a given assignment parameter, we now restrict our analysis to systems operating under these optimal partitionings. Our goal in this section is to characterize the optimal assignment parameter $p^*(\lambda)$, defined as:

$$p^*(\lambda) = \min_{p \in \mathcal{P}_\lambda} T^*_{\text{DM}}(p;\lambda)$$

which fully specifies the optimal operating point.

Solving for the critical points of $T^*_{\text{DM}}(\cdot;\lambda)$ analytically involves solving an eighth-degree polynomial equation, which generally does not yield a simple closed-form solution and typically requires numerical optimization techniques. However, leveraging insights from our preliminary analysis, we identify scenarios where the optimization problem simplifies.

A. Redundancy of Local-Heavy Mode

In this subsection, we characterize a family of dual-mode systems in which the "local-heavy" SM2 is redundant and the optimal strategy is to assign all jobs to SM1. We formalize this result below.

Theorem 4 (Redundant Mode). For a dual-mode system, if the canonical parameter f_1 of Service Mode 1 satisfies:

$$f_1 \geq \frac{1}{K+1}$$

then, the delay optimal assignment parameter,

$$p^*(\lambda) = 1$$

for all $\lambda \in \Lambda_{\text{DM}}$.

Proof: Proof deferred to technical report [**?**]. ■

B. Breakaway Behavior of Optimal Assignment

We now consider the complementary case $f_1 < 1/(K+1)$, which includes the class of *throughput-efficient* systems (Definition 1). In this regime, the optimal assignment parameter $p^*(\lambda)$ exhibits a breakaway structure: At low loads, it is optimal to exclusively assign jobs to the "cloud-heavy" SM1 ($p^*(\lambda) = 1$), whereas at higher loads the optimal policy breaks away from exclusive SM1 assignment and allocates some jobs to SM2 ($p^*(\lambda) \in [0, 1)$).

Theorem 5 (SM1 Optimality at Low Loads). For a dual-mode system, if the parameter f_1 of Service Mode 1 satisfies:

$$f_1 < \frac{1}{K+1}$$

Then, the delay optimal assignment parameter,

$$p^*(\lambda) = 1$$

for all $\lambda \in \Lambda_{\text{DM}}$ such that $\lambda \leq \frac{\mu_0(K-\sqrt{K})}{1-f_1(1+\sqrt{K})}$.

Proof: For all arrival rates $\lambda \leq \frac{\mu_0(K-\sqrt{K})}{1-f_1(1+\sqrt{K})}$, we have $f^*(\lambda) \leq f_1 < f_2$. The remainder follows from the proof of Theorem 4. ∎

We next show that at sufficiently high loads, the optimal policy departs from exclusive SM1 assignment.

Theorem 6 (Break away from SM1 at High Loads). For a dual-mode system, if the parameter f_1 of Service Mode 1 satisfies:

$$f_1 < \frac{1}{K+1}$$

Then, the delay optimal assignment parameter,

$$p^*(\lambda) \in [0, 1)$$

for all arrival rates, $\lambda \in \Lambda_{\text{DM}}$ such that:

$$\lambda \geq \frac{K\mu_0}{1-f_1}$$

Proof: The result follows from stability considerations. Full details are omitted for brevity. ∎

We now characterize the high load behavior of the system in the following subsection.

C. High Load Regime

In the high-load regime, the optimal assignment parameter $p^*(\lambda)$ may exhibit non-monotonic behavior. For systems with $f_1 < 1/(K+1)$, this behavior is governed by the canonical parameter f_2. We distinguish two cases: (i) *throughput-efficient* systems with $f_2 > 1/(K+1)$, and (ii) systems with $f_2 \leq 1/(K+1)$, where SM2 becomes a throughput bottleneck.

We first establish that, for throughput-efficient systems, the optimal strategy in the high-load regime involves assigning jobs to both service modes:

Theorem 7. For a *throughput-efficient* dual-mode system, the optimal assignment parameter satisfies:

$$p^*(\lambda) \in (0, 1)$$

for all $\lambda \in \Lambda_{\text{DM}}$ such that $\lambda > \max\left\{\frac{\mu_0}{f_2}, \frac{K\mu_0}{1-f_1}\right\}$. Moreover, in the limit of full system utilization, the optimal assignment converges to

$$\lim_{\lambda \to \lambda_{\max}^-} p^*(\lambda) \to \frac{f_2 - 1/(K+1)}{f_2 - f_1}$$

where $\lambda_{\max} = (K+1)\mu_0$.

Proof: Proof deferred to the technical report [**?**]. ∎

We now consider the second scenario, establishing that for sufficiently high loads, exclusive assignment to SM2 becomes optimal:

Theorem 8. For a dual-mode system, if the parameter f_2 of Service Mode 1 satisfies:

$$f_2 \leq \frac{1}{K+1}$$

Then, the delay-optimal assignment parameter satisfies:

$$p^*(\lambda) = 0$$

for all arrival rates, $\lambda \in \Lambda_{\text{DM}}$ such that $\lambda \geq \frac{\mu_0(K-\sqrt{K})}{1-f_2(1+\sqrt{K})}$.

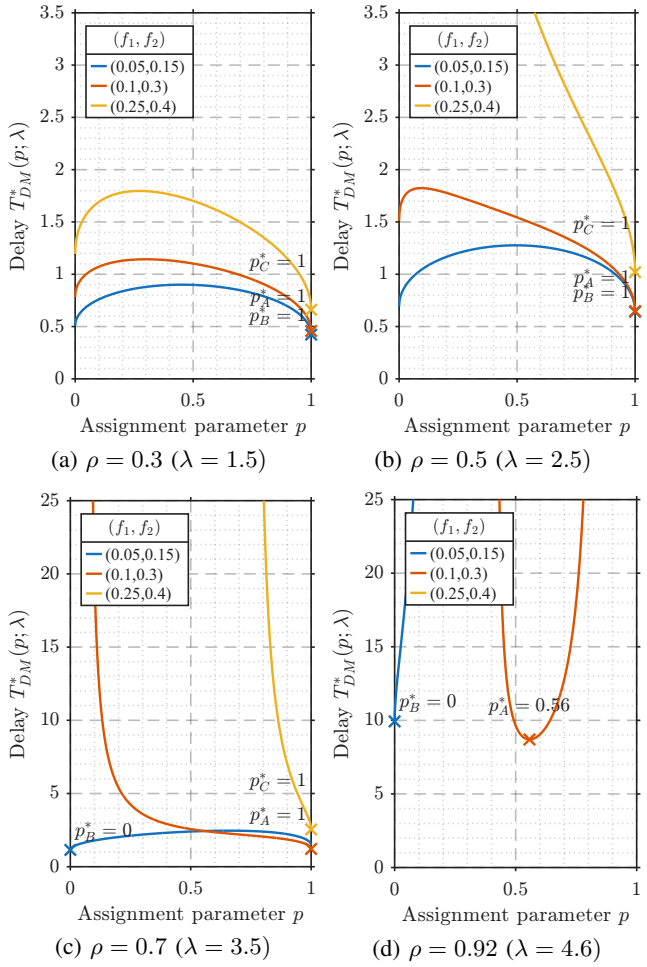
Proof: For arrival rates $\lambda \geq \frac{\mu_0(K-\sqrt{K})}{1-f_2(1+\sqrt{K})}$, we have $f_1 < f_2 \leq f^*(\lambda)$. The remainder follows from the proof of Theorem 4. ∎

VII. Performance Evaluation

In this section, we illustrate the analytical insights derived in earlier sections using numerical computation. We focus our investigation on three representative dual-mode systems, each defined by distinct canonical service mode parameters. To facilitate meaningful comparisons, we normalize the local server capacity to $\mu_0 = 1$ and set the cloud-to-local capacity ratio to $K = 4$, resulting in a total system capacity of $\mu^* = (K+1)\mu_0 = 5$. The arrival rate is represented as a normalized load parameter $\rho = \lambda/\mu^*$, signifying the fraction of total system capacity utilized.

The three systems considered are: System A, a throughput-efficient configuration with $f_1 = 0.1$ and $f_2 = 0.3$; System B, a configuration overly reliant on cloud processing, with $f_1 = 0.05$ and $f_2 = 0.15$; and System C, which is overly reliant on local resources, with $f_1 = 0.25$ and $f_2 = 0.4$.

As Systems B and C are not throughput-efficient, they cannot support the full arrival rate up to μ^*. Using Theorem 2,



(a) $\rho = 0.3$ ($\lambda = 1.5$) (b) $\rho = 0.5$ ($\lambda = 2.5$)

(c) $\rho = 0.7$ ($\lambda = 3.5$) (d) $\rho = 0.92$ ($\lambda = 4.6$)

Fig. 4. Delay $T_{\text{DM}}(p;\lambda)$ vs. assignment parameter p under optimal partitioning, shown for increasing system loads. p^*s indicate the optimal assignment parameter.

we determine the maximum stabilizable loads: $\rho^A_{\max} = 1$, $\rho^B_{\max} = 0.94$, and $\rho^C_{\max} = 0.8$.

A. Benefit of Optimal Assignment

In Figure 4, we plot the delay of each system, denoted by $T^*_{\text{DM}}(p;\lambda)$, as a function of the assignment probability p under optimal server partitioning. Subfigures (a) through (d) represent different load conditions to demonstrate the sensitivity of delay performance to suboptimal assignment choices across feasible values of p.

Consistent with our analysis, at low to moderate loads (Figures 4a-b), all systems achieve optimal delay performance with exclusive SM1 assignment ($p^* = 1$). As loads increase (Figure 4c), the feasible range of p significantly diminishes in System C, revealing a stability bottleneck due to excessive reliance on local processing. Notably, System B has better delay performance compared to the throughput-efficient System A for a substantial range of p up to a load of about 0.7.

Approaching system capacity (Figure 4d), System A achieves better performance under optimal assignment as

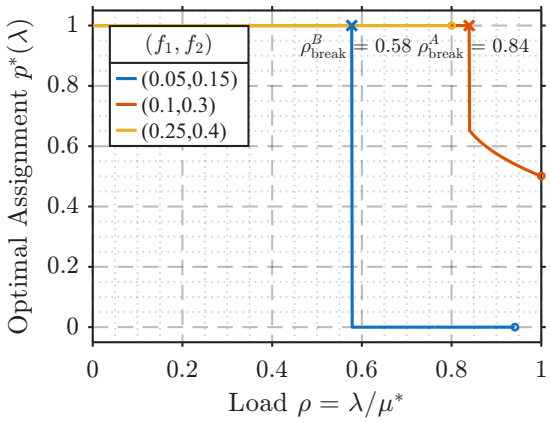


Fig. 5. Optimal assignment $p^*(\lambda)$ vs. load ρ. Systems A and B exhibit breakaway transitions at ρ^A_{break} and ρ^B_{break}. System C remains at $p^* = 1$.

System B's cloud queues become increasingly congested due to its cloud reliance.

Finally, we observe that optimal selection of the assignment parameter is crucial at higher loads, where small deviations about the optimal operating point result in relatively larger delay penalties.

B. Assignment Dynamics Across Load

In Figure 5, we plot the evolution of the optimal assignment parameter $p^*(\lambda)$ as a function of system load and verify the structural behaviors predicted in our analysis. Note that the curves for System B and C extend up to the maximum stabilizable load of 0.94 and 0.8 respectively.

Consistent with our analysis, System C which belongs to the family of redundant mode systems described in Theorem 4 always operates under exclusive SM1 assignment ($p^* = 1$) across all feasible loads. In contrast, Systems A and B exhibit a breakaway structure, transitioning from exclusive SM1 assignment at lower loads ($p^* = 1$) to mixed or exclusive SM2 assignment at higher loads. We explicitly highlight the breakaway points, ρ^A_{break} and ρ^B_{break}, demonstrating these load-dependent transitions.

As loads approach system capacity, the optimal assignment in System A converges to $\frac{f_2 - 1/(K+1)}{f_2 - f_1} = 0.5$, as indicated by Theorem 7. Similarly, at higher loads, System B operates under optimal assignment of $p^* = 0$, as established in Theorem 8.

C. Delay under Optimal Assignment

Figure 6 compares the delay performance of dual-mode systems operating with the optimal assignment parameter, $T^*_{\text{DM}}(p^*(\lambda);\lambda)$, against system load. We also include the delay performance of a tunable-mode system operating at its optimal service fraction, which serves as a fundamental lower bound for the dual-mode systems (Proposition 1).

At the breakaway loads ($\rho^A_{\text{break}}, \rho^B_{\text{break}}$), we note a discontinuity in the slope of the delay curve, reflecting changes in assignment strategies.

System C performs worst across all loads, consistent with our analysis: the tunable-mode optimum is cloud-reliant,

whereas both of System Cs modes are local-heavy ($f^*(\lambda) < 1/(K+1) < f_1 < f_2$).

In Figure 6b, we highlight points where the delay curves for Systems A and B intersect the tunable-mode lower bound at loads $\rho^A_{\text{touch}} = 0.57$, $\rho^B_{\text{touch1}} = 0.47$, and $\rho^B_{\text{touch2}} = 0.73$. As established in our analysis, the optimal service fraction in the tunable-mode system, $f^*(\lambda)$, increases monotonically from 0 at low loads to $1/(K+1)$ as the system approaches full utilization. These intersections occur at the loads where $f^*(\lambda)$ matches the canonical service mode parameters: f_1 for System A, and f_1 or f_2 for System B.

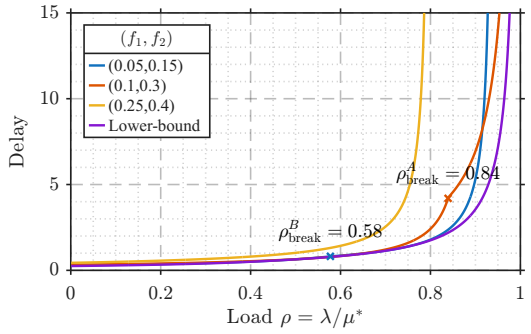
At lower loads, System B generally outperforms System A. However, due to its reliance on cloud-heavy modes, System B is unable to stabilize near full capacity, leading to substantially higher delays in the high-load regime. This phenomenon underscores a critical system design trade-off: Throughput-efficient service mode design ($f_1 < 1/(K+1) < f_2$) allows fully utilizing the servers' capacities and maximizes the stability region. However, systems that are not throughput efficient, with ($f_1 < f_2 < 1/(K+1)$) can outperform the throughput-efficient system at lower loads while sacrificing the ability to stabilize near capacity.

VIII. Conclusion

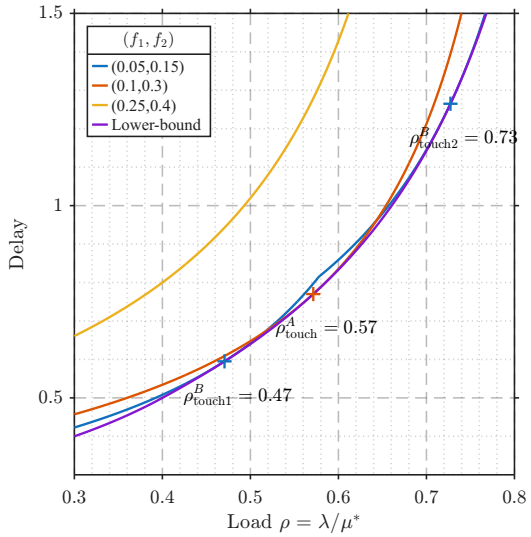
We studied a computation offloading system consisting of a sequential local and cloud server, supporting two distinct service modes differentiated by how processing tasks are divided between these servers. We characterized the system's stability region and derived principles for designing throughput-efficient service modes.

We then derived the optimal server resource allocation for a fixed job assignment strategy and presented a compact expression for delay under these optimal partitionings. This representation allowed us to identify a fundamental lower bound on delay and interpret the dual-mode system by comparison to a tunable single-mode system.

Finally, we showed that under optimal resource allocation, the delay-optimal assignment strategy exhibits a breakaway structure: at lower loads, all jobs are assigned to the cloud-heavy mode, while at higher loads, the system breaks away from this strategy and assigns some or all jobs to the local-heavy mode. Our simulation results confirm and illustrate these theoretical insights.



(a) Delay under optimal assignment $p^*(\lambda)$ vs. load ρ. Break points mark shifts in assignment strategy.



(b) Zoomed-in view highlighting loads where Systems A and B match the tunable-mode lower bound at ρ^A_{touch}, ρ^B_{touch1}, and ρ^B_{touch2}.

Fig. 6. Delay performance under optimal assignment compared to the tunable-mode lower bound. Subfigure (b) highlights points where the bound is achieved.

References

[1] X. Jiang, Y. Zhou, S. Cao, I. Stoica, and M. Yu, "Neo: Saving gpu memory crisis with cpu offloading for online llm inference," 2024, preprint published Nov 2, 2024.

[2] Z. Hao, H. Jiang, S. Jiang, J. Ren, and T. Cao, "Hybrid slm and llm for edge-cloud collaborative inference," ser. EdgeFM '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 36–41.

[3] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, and L. Tang, "Neurosurgeon: Collaborative intelligence between the cloud and mobile edge," ser. ASPLOS '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 615–629.

[4] X. Qin, B. Li, and L. Ying, "Efficient distributed threshold-based offloading for large-scale mobile cloud computing," *IEEE/ACM Transactions on Networking*, vol. 31, no. 1, pp. 308–321, 2023.

[5] X. Qin, Q. Xie, and B. Li, "Distributed threshold-based offloading for heterogeneous mobile edge computing," in *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, 2023, pp. 202–213.

[6] J. Zhou, D. Tian, Z. Sheng, X. Duan, and X. Shen, "Distributed task offloading optimization with queueing dynamics in multiagent mobile-edge computing networks," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 311–12 328, 2021.

[7] H. Feng, J. Llorca, A. M. Tulino, and A. F. Molisch, "Optimal dynamic cloud network control," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2118–2131, 2018.

[8] Y. Cai, J. Llorca, A. M. Tulino, and A. F. Molisch, "Mobile edge computing network control: Tradeoff between delay and cost," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020, pp. 1–6.

[9] D. Jeff and E. Modiano, "Optimal service mode assignment in a simple computation offloading system." Allerton Conference on Communication, Control, and Computing, 2025.

[10] D. P. Bertsekas and R. G. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1992.

[11] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*. Morgan & Claypool Publishers, 2010, vol. 1.